

VŠB – TECHNICKÁ UNIVERZITA OSTRAVA

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

VŠB – TECHNICKÁ UNIVERZITA OSTRAVA

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEDRA INFORMATIKY

**Absolvování individuální odborné praxe**

**Individual Professional Practice in the Company**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student:

**Patrik Helia**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe  
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: SCOVECO, s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Mgr. Jiří Dvorský, Ph.D.**

Konzultant bakalářské práce: Ing. Zdeněk Velart, Ph.D.

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.



.....

Patrik Helia  
30. dubna 2014

## **Poděkování**

Rád bych poděkoval společnosti SCOVECO s.r.o. za to, že mi umožnila vykonávat odbornou praxi v této společnosti a jejím zaměstnancům, kteří mi byli oporou při jejím absolvování.

## **Prohlášení zástupce spolupracující právnické nebo fyzické osoby**

„Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“



.....  
podpis zástupce  
30. dubna 2014

## **Abstrakt**

Tato bakalářská práce pojednává o průběhu mé bakalářské praxe ve společnosti SCOVECO s.r.o., která si klade za cíl prosazovat jednoduchá a inovativní řešení. Praxi jsem vykonával pod svým vedoucím Ing. Zdeňkem Velartem, Ph.D. Při vykonávání jsem pracoval na různých projektech.

## **Klíčová slova**

Yii, Framework, HTML, jQuery, CSS, PHP, MySQL, Javascript; Java, Hibernate, Eclipse BIRT

## **Abstract**

This bachelor thesis deals with course of my Bachelor practice at SCOVECO company, which aims to promote simple and innovative solutions. Practice I performed under my leading Ing. Zdenek Velart, Ph.D. When exercising, I worked on various projects.

## **Keywords**

Yii, Framework, HTML, jQuery, CSS, PHP, MySQL, Javascript; Java, Hibernate, Eclipse BIRT



## Seznam použitých zkratek

Zkratka	Význam
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>CSS</b>	Cascading Style Sheets
<b>CSV</b>	Comma-separated values
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>MVC</b>	Model-View-Controller
<b>MySQL</b>	Multiplatformní databáze
<b>PDF</b>	Portable Document Format
<b>PHP</b>	Hypertext Preprocessor
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>XML</b>	Extensible Markup Language

# Obsah

Obsah.....	10
Úvod.....	11
Pracovní zařazení.....	11
1    Zadané úkoly.....	12
1.1    Online rezervační systém.....	12
1.2    Informační systém pro síť kaváren.....	12
1.3    Informační systém pro správu událostí.....	12
2    Rezervační systém.....	14
2.1    Řešení jednotlivých problémů.....	14
2.2    Časová náročnost.....	16
3    Informační systém pro síť kaváren.....	17
3.1    Řešení jednotlivých problémů.....	17
3.2    Časová náročnost.....	20
4    Informační systém pro správu událostí.....	21
4.1    Řešení jednotlivých problémů.....	21
4.2    Časová náročnost.....	23
5    Nabyté znalosti.....	24
6    Závěr.....	25

# Úvod

Cílem této bakalářské práce je představit firmu SCOVECO s.r.o. a shrnout průběh individuální odborné praxe, kterou jsem v této firmě vykonával. Budu popisovat vybrané úkoly, na kterých jsem v průběhu praxe pracoval, uvedu postup při jejich řešení a detailněji rozvedu problematiku hlavních částí řešených úkolů.

## Profil společnosti SCOVECO s.r.o.

SCOVECO, s.r.o. je zkušená dynamická společnost. Byla založena v roce 2011 jako firma, která si klade za cíl prosazovat jednoduchá a inovativní řešení jednoduchých i komplexních problémů pomocí moderních IT technologií. Záměrem společnosti je budovat, prosazovat a nasazovat IT řešení, která opravdu usnadňují život firmám i lidem. [1]

## Pracovní zařazení

Po kontaktování společnosti SCOVECO, s.r.o. jsem byl pozván na vstupní pohovor, který proběhl formou ústní konzultace, kde mi bylo zároveň sděleno moje přijetí na odbornou praxi, kde jsem nastoupil 29. září na pozici Developera.

# 1 Zadané úkoly

V průběhu odborné praxe jsem pracoval na různých typech úkolů zahrnujících analýzu, návrh, vývoj a optimalizaci aplikací. Především jsem řešil programátorské úlohy v jazyce PHP a Java. Většinu zadaných úkolů jsem řešil samostatně a některé pak ve spolupráci s mým vedoucím praxe či kolegy.

## 1.1 Online rezervační systém

Jako jeden z prvních úkolů ve firmě jsem dostal navrhnout a naimplementovat systém pro online rezervace na výukové kurzy a festivaly. Při navrhování systému jsem se musel vypořádat s dnešní době již neobvyklými požadavky ze strany klienta. Jelikož na svém webhostingu neměl zaplacenou žádnou databázi, byli jsme nuceni veškeré informace ukládat do souborů. Zákazník měl na systém další požadavky jako generování faktur do PDF a možnost odesílat emaily.

## 1.2 Informační systém pro síť kaváren

Hlavní projekt společnosti SCOVECO s.r.o. byl vývoj informačního systému pro síť provozoven. Tento informační systém je vyvíjen ve frameworku Eclipse Scout, který je založený na technologii Java. Výhody tohoto frameworku spočívají v tom, že vyvíjíte jednu aplikaci, která slouží jako desktopová aplikace, aplikace pro mobilní zařízení a webová aplikace.

Pro spojení s databází se využívá známý framework Hibernate, který vyvíjí společnost JBoss. Je to framework umožňující objektově relační mapování, což znamená, že mapuje objekty na entity v relační databázi. Další úkol, který mi byl zadán na tomto projektu bylo, seznámit se, a naučit se pracovat s reportovacím nástrojem Eclipse BIRT, který slouží k vytváření šablon, ze kterých se následně budou generovat reporty

## 1.3 Informační systém pro správu událostí

Jako poslední úkol jsem dostal analyzovat, navrhnout a následně také implementovat systém pro online přihlašování na různé akce, události a semináře. Podmínkou bylo, aby šlo vše ovládat skrze online administraci. Uživatelé si mohli vytvářet účty, a následně je spravovat. Dále bylo nutné implementovat možnosti generování PDF podle předpřipravených šablon, odesílání emailu s fakturami a pro zapomenuté hesla. Kvůli rozsáhlosti systému jsem se rozhodl použít poměrně mladý, ale výkonný open-source striktně objektově orientovaný MVC framework Yii, který je určen pro vývoj rozsáhlých aplikací.

MVC vychází z třívrstvé architektury, rozděluje aplikaci do tří na sobě nezávislých vrstev. Datová vrstva, aplikační vrstva a prezentační vrstva. Datová vrstva reprezentuje data a přístup k datům.

Prezentační vrstva převádí data reprezentované modelem do podoby vhodné pro prezentaci uživateli a následné předání požadavků od uživatele k zpracování. Aplikační vrstva reaguje na události vyvolané uživatelem a zajišťuje změny v modelu nebo pohledu. [2]

## 2 Rezervační systém

### 2.1 Řešení jednotlivých problémů

V prvním systému nám šlo o to, aby se zákazník mohl přihlásit přes online formulář na vzdělávací kurzy. Jako první jsme navrhli vzhled formuláře, přes které se realizují všechny akce viz. Obrázek 2.1. Vize byla jednoduchá, po volbě online přihlášky si bude možnost vybrat, zda máte zájem o jednodenní kurzy nebo festivaly. Po vybrání typu budeme přesměrování na přihlašovací formulář, kde systém načte data ze souboru CSV. CSV je jednoduchý souborový formát pro úschovu tabulkových dat, která jsou oddělena středníkem. Stanovili jsme jednotný formát vložených dat s tím, že pokud hodnota nebude vyplněna, bude reprezentována hodnotou NULL nebo prázdným řetězcem.

#### Struktura CSV souboru:

```
P;22;28.09.2013; null; Perspektiva I 17.00 - 19.00 přednáška (2hod);  
300; 300; 0; 14, 10-09-2012;null;null; Ostrava
```

Jelikož každý formulář obsahoval různé prvky, bylo nutné jako první atribut mít jestli se jedná o festival nebo jednodenní kurz. Pro rozhodnutí které budeme vybírat slouží parametr, který se předává po odeslání http požadavku pomocí proměnné GET. Jako druhý atribut je stanovený číslo kurzu, poté následuje začátek a konec kurzu ve formátu den-měsíc-rok. Pokud není konec kurzu stanoven nebo je nastaven na NULL, tak se počítá s tím, že kurz začíná i končí stejný den. Další atributy nám udávají název akce, cenu, cenu za ubytování, zálohu, den splatnosti a poslední datum registrace, od jakého věku je možné se na kurz registrovat, do jakého věku je možné se na kurz registrovat a místo konání.

Pro parsování CSV jsem použil PHP funkce *fopen* která přijímá dva parametry, první je cesta k souboru a druhý je mód jakým bude otevřen. Pro můj účel jsem použil pouze mód pro čtení, jelikož nemáme v plánu nic zapisovat. Druhá funkce je *fgetcsv*, která prochází soubor po řádcích, které následně rozdělí podle zadaného oddělovače a vrátí pole hodnot. Jelikož si zákazník přál, aby návštěvníkům bylo umožněno přihlásit se na více kurzů najednou, tak jsem se rozhodl vytvořit si jednoduchou funkci v javascriptu, která vkládá pomocí jQuery výběrací box s inkrementovaným Id, který se inkrementoval i ve skrytém textovém poli, to mi umožnilo vybrané elementy také mazat

FESTIVALY		Kurzy	
Kurz	<input style="width: 90%;" type="text" value="Text"/>		
Ubytování	<input style="width: 90%;" type="text" value="Text"/>		
<input checked="" type="checkbox"/>	Mám zájem o ubytování		
Účastník akce		Rodič(u účastníků do 18-let)	
Jméno*	<input style="width: 90%;" type="text" value="Zadat text"/>	Jméno*	<input style="width: 90%;" type="text" value="Zadat text"/>
Příjmení*	<input style="width: 90%;" type="text" value="Zadat text"/>	Příjmení*	<input style="width: 90%;" type="text" value="Zadat text"/>
Datum narození*	<input style="width: 90%;" type="text" value="Zadat text"/>	Datum narození*	<input style="width: 90%;" type="text" value="Zadat text"/>
Telefon*	<input style="width: 90%;" type="text" value="Zadat text"/>	Telefon*	<input style="width: 90%;" type="text" value="Zadat text"/>
Email*	<input style="width: 90%;" type="text" value="Zadat text"/>	Email*	<input style="width: 90%;" type="text" value="Zadat text"/>
Adresa účastníka		<input checked="" type="checkbox"/> Fakturační údaje(firma)	
Ulice*	<input style="width: 90%;" type="text" value="Zadat text"/>	Společnost*	<input style="width: 90%;" type="text" value="Zadat text"/>
Číslo popisné*	<input style="width: 90%;" type="text" value="Zadat text"/>	Jméno*	<input style="width: 90%;" type="text" value="Zadat text"/>
Město*	<input style="width: 90%;" type="text" value="Zadat text"/>	Příjmení*	<input style="width: 90%;" type="text" value="Zadat text"/>
Psč*	<input style="width: 90%;" type="text" value="Zadat text"/>	Město*	<input style="width: 90%;" type="text" value="Zadat text"/>
Stát*	<input style="width: 90%;" type="text" value="Zadat text"/>	Ulice*	<input style="width: 90%;" type="text" value="Zadat text"/>
		Číslo popisné*	<input style="width: 90%;" type="text" value="Zadat text"/>
		Psč*	<input style="width: 90%;" type="text" value="Zadat text"/>
		Stát*	<input style="width: 90%;" type="text" value="Zadat text"/>
		IČ*	<input style="width: 90%;" type="text" value="Zadat text"/>
		DIČ	<input style="width: 90%;" type="text" value="Zadat text"/>
Cena kurzu			
Cena výuky	<input style="width: 90%;" type="text" value="Zadat text"/>		
Cena ubytování	<input style="width: 90%;" type="text" value="Zadat text"/>		
Cena celkem	<input style="width: 90%;" type="text" value="Zadat text"/>		
Záloha	<input style="width: 90%;" type="text" value="Zadat text"/>		
Doplatek	<input style="width: 90%;" type="text" value="Zadat text"/>		
<input checked="" type="checkbox"/> Prohlašuji že souhlasím s obchodními podmínkami			
Ochrana proti spamu		Opište bezpečnostní kod	
<div style="border: 1px solid black; padding: 5px; text-align: center;">ABCDE</div>		<input style="width: 90%;" type="text" value="Zadat text"/>	
<input type="button" value="Odeslat"/>			

Obrázek 2.1: Návrh formuláře

Pro zákazníka je také důležité, aby byl informovaný o aktuální ceně jím vybraných kurzů, proto jsem se rozhodl, že po každé akci, která ovlivní cenu se zavolá javascriptová funkce na přepočítání aktuální ceny. Jelikož pracuje javascript na straně klienta, bylo také nutné vše přepočítávat na straně serveru, abychom zajistili, že se k nám nedostanou podvržené data.

Pro účastníky mladší 18 let je nutné, aby se u přihlašování evidovali i rodiče. Proto se při vyplňování data narození účastníka kontroluje věk pomocí javascriptu. Pokud je věk menší než 18 let, tak se pomocí jQuery zviditelní druhá část formuláře pro rodiče. Po vyplnění všech údajů a odeslání proběhne na serveru validace dat od účastníka a informace jsou uloženy do souboru.

Pokud proběhne validace korektně, tak se spustí generování PDF souboru. Pro generování PDF jsem si zvolil knihovnu *html2pdf*, která z html šablony dokáže vygenerovat PDF soubor. Při generování jsem se ze začátku potýkal se špatným formátováním českých znaků, které jsem musel řešit vložením sady fontů přímo do knihovny a následně fonty embedovat do PDF souboru, tím jsem zaručil správné zobrazování českých znaků.

Jelikož nebyl na zákaznickově webhostingu zprovozněn smtp server, musel jsem vymyslet náhradní řešení. Rozhodl jsem se, že využiji smtp server od společnosti Google, Inc. Proto jsem zvolil knihovnu PHPMailer, u které lze jednoduše nakonfigurovat adresu, port serveru, přihlašovací jméno a heslo. Další výhody, která tato knihovna poskytuje je možnost přidání více příjemců. Jelikož se zpočátku obsah emailové zprávy udával přímo v zdrojovém kódu aplikace, tak zákazník po první schůzce měl požadavek, aby se obsah načítal z šablony, kterou by mohl sám lehce editovat, bez nutnosti zasahovat do zdrojového kódu.

## 2.2 Časová náročnost

**Tabulka 2.1:** Časová náročnost projektu

Fáze projektu	Časová náročnost
Analýza a návrh řešení	2 Dny
Vytváření formuláře a stylizace	1 Den
Základní funkčnost	3 Dny
Implementace javascriptu a jQuery	3 Dny
Tvorba šablon pro PDF, generování PDF	3 Dny
Odesílání emailu	2 Dny
Testování aplikace a optimalizace	3 Dny



## 3 Informační systém pro síť kaváren

### 3.1 Řešení jednotlivých problémů

Po ukončení práce na prvním projektu jsem dostal za úkol seznámit se s Eclipse BIRT, který slouží k vytváření reportů. Jelikož jsem neměl žádné předchozí zkušenosti s žádným podobným nástrojem, bylo nutné se nejprve zorientovat ve vývojovém prostředí. K tomu mi posloužily tutoriály, které jsou oficiálních stránkách Eclipse BIRT.

Pro správný chod Eclipse BIRT je nutné k reportu přidávat také data. To lze dvěma způsoby. Jako první se lze připojit přímo na MySQL databázi. Při druhém způsobu se v reportu nastaví cesta k XML. V našem projektu se k připojování dat používá souboru XML.

FreeMarker je šablonovací engine pro Java technologie. Slouží ke generování textových výstupů ze šablon. Tento balíček není určen pro koncového uživatele, ale slouží pro programátory, kteří jej mohou vložit do svých produktů. [3] Moje práce spočívala ve vytvoření testovacího XML souboru s definovanou strukturou a testovacími daty, který sloužil pro tvorbu reportu. Poté vytvořit samotný report tak, aby zobrazoval dle definovaného zadání jak texty s přesně určenou pozicí, statických tabulek, tak také dynamicky generovaných tabulek (cross-table). V poslední řadě jsem musel vytvořit šablonu pro FreeMarker tak, aby po naplnění daty vzniklo XML s mojí předem definovanou strukturou. Data pro šablonu mi připravoval spolupracovník.

Jako první jsem vytvářel „Produkční plán“, který eviduje na který den se plán vztahuje, Pro jakou směnu patří, kteří pracovníci budou na směně pracovat. Poté se generují tabulky s informacemi o tom, kolik je potřeba vyrobit danou směnu určitého zboží. Při vytváření reportu bylo třeba vytvořit tabulku, u které se budou automaticky generovat sloupce tabulky podle dat. Jelikož u elementu tabulky lze vkládat sloupce pouze staticky, použil jsem element cross-table, u které jde automaticky generovat názvy sloupců, příklad uvádím na obrázku 3. 1 a obrázku 3.2. K tomuto účelu jsem musel definovat v rámci reportu datový zdroj Data cube, který umožňuje určit, která data budou tvořit sloupce dynamicky generované tabulky a která jako řádky. BIRT na základě této definice automaticky provede grupování dat.

[name]		
Položka		[premise]
	[article]	Σ[count_Group1/art...]
Cross Tab		Footer Row

**Obrázek 3. 1:** *Cross-table v Eclipse BIRT*

Dezerty					
Položka	Sloupec1	Sloupec2	Sloupec3	Sloupec4	Sloupec5
Položka č.1	52.5	0	52		

**Obrázek 3.2:** *Cross-table v PDF souboru*

Další report, který jsem vytvářel, byl „Dodací list“. Tento report se generoval vždy pro jednu pobočku. Zde se generovalo unikátní číslo dodacího listu, název zboží, cena, stav na pobočce z předchozího dne, kolik kusů se bude daný den dovážet a kolik bude na pobočce zboží celkově. Problém spočíval u zboží, které mělo omezenou délku trvanlivosti. Zde bylo potřeba generovat další sloupce do tabulky tak, aby bylo možné evidovat kolik, zbývá zboží do vypršení trvanlivosti. Jelikož Eclipse BIRT nedokázal nejprve spočítat jakou má výrobek trvanlivost, musel jsem tomu uzpůsobit formát XML tak, aby šlo jednoznačně určit, kolik sloupců se má vygenerovat. Jelikož jsem věděl, že žádné zboží nemá delší trvanlivost než 5 dní, tak jsem do XML přidal atribut, který nabývá hodnot 0 nebo -1, pokud je hodnota -1, tak se údaj nebude zobrazovat v tabulce, pokud je hodnota 0, tak údaj bude zobrazen i v tabulce viz obrázek 3.3, kde je zobrazena zmíněná tabulka v reportu a obrázek 3.4, kde je výsledek v PDF.

Položka	Cena	Den před	Závoz	Akt.stav	Stav večer						Zůstatek	Odpis			
<div><div></div><div></div>Detail</div>															
[articleName]					Dneš	2D-R	2.den	3D-R	3.den	4D-R	4.den	5D-R	5.den	[dayCount]	
<div><div></div><div></div>Grid</div>															
[article]	[pnice]	[eveni]	[distri]	[sum]		[dist]	[day2]	[dist]	[day3]	[dist]	[day4]	[dist]	[day5]	[balance]	[writeOff]

**Obrázek 3.3:** *Ukázka z šablony Eclipse BIRT*

Položka	Cena	Den před	Závoz	Akt.stav	Stav večer										Zůstatek	Odpis
Kategorie					Dnešní	2D-R	2.den	3D-R	3.den	4D-R	4.den	5D-R	5.den			
Položka 1	00.00	0	10	15												
Položka 2	00.00	0	10	15		0										
Položka 3	00.00	0	10	15		0		0								
Položka 4	00.00	0	10	15		2		3		0						
Položka 5	00.00	0	10	15		4		5		0		0				

**Obrázek 3.4:** *Výsledek*

Poslední report, který jsem vyvíjel, byl "Objednávka zboží" u subdodavatelů. Po předchozích zkušenostech byl tento report lehký na vývoj, zde se používaly jenom tabulky, které byly naplněny informacemi o subdodavateli, zákazníkovi, dodací adresa na kterou má objednané zboží být doručeno, objednané zboží a v poslední řadě počet kusů. Později se tento report generoval do PDF, které bylo následně odesíláno emailem.

Po dokončení všech potřebných reportů jsem byl zaškolen do vývoje informačního systému. Jelikož systém vyvíjelo více spolupracovníků, bylo potřeba systém nějakým způsobem verzovat. K tomu bylo využíváno verzovacího systému Git. Git je volně šiřitelný open-source distribuovaný systém pro správu verzí navržený tak, aby bylo lehké manipulovat jak s malými tak s velkými projekty rychle a efektivně. [4] Pro seznámení mi bylo doporučeno si projít dokumentaci programu a vyzkoušet si tutoriál, který nám nabízí na oficiálních stránkách projektu git.

Za pomoci gitu jsem si stáhl repositář s aktuální verzí vyvíjeného systému. Framework Eclipse Scout je hodně rozsáhlý a komplikovaný, proto bylo nezbytné naučit se zpočátku ovládat základní prvky. Můj první úkol bylo upravit kalendář tak, aby z něj bylo jednoznačně na první pohled rozeznat, v jakém stavu je daná kategorie produktů. Při tomto úkolu bylo nutné pracovat i s databází, se kterou pracoval framework hibernate. Hibernate je framework napsaný v jazyce Java, který umožňuje tzv. objektově-relační mapování (ORM). Usnadňuje řešení otázky zachování dat objektů i po ukončení běhu aplikace. [5] Tento framework přistupuje do databáze pomocí takzvaných dao tříd, které již byly v projektu vytvořeny. Zde jsem si musel vytvořit metody, které mi z databáze budou vracet výsledky o jednotlivých stavech. Pote jsem do formuláře, který reprezentoval kalendář, doimplementoval rozdělení jednotlivých oken na 6 částí, kde každá část reprezentovala jeden typ zboží. K uzavření denní uzávěrky bylo potřeba mít všechny typy zboží naplánovány a kalendář tak slouží k rychlému přehledu.

## 3.2 Časová náročnost

**Tabulka 3.1:** *Časová náročnost projektu*

Fáze projektu	Časová náročnost
Výuka v Eclipse BIRT	4 Dny
Report „Dodací list“	3 Den
Report „Produkční plán“	5 Dny
Report „Objednávka zboží“	1 Den
Seznámení s Eclipse Scout	2 Dny
Úprava kalendáře	1 Den
Odesílání emailů subdodavatelům	1 Dny

## 4 Informační systém pro správu událostí

### 4.1 Řešení jednotlivých problémů

Další projekt jsem dostal navrhnout a naprogramovat modul, který bude následně jednoduchý naimplementovat do již stávajícího webu. Tento modul bude sloužit pro správu různých akcí, na které se lze online přihlašovat. Bude možné dynamicky přidávat různé kategorie. Do každé kategorie lze vkládat akce, u kterých lze nastavit kritéria jako dostupnost od a do zadaného věku, maximální počet účastníků. Dále bude možné vytvořit pro každou akci doplňkové služby jako například možnosti ubytování. Při přihlašování bude možné vytvořit uživatelský účet, na kterém bude uživatel moci najít přehled všech svých objednávek. Pokud bude uživatel přihlášený jako administrátor, bude moci spravovat vše z webového prostředí.

Další požadavek byl ověřování a validace dat již na straně klienta. K tomu jsem se rozhodl používat technologii Ajax, pomocí které zjišťuji, zda je již zadané uživatelské jméno v databázi nebo je volné. Tuto technologii také používám k zobrazení souhrnu informací před odesláním formuláře.

Po analýze požadavků jsem navrhl model databáze s tabulkami kategorie, ve které eviduji pouze jméno kategorie. Dále jsem modeloval tabulku kurzů, která bude spadat vždy pod jednu kategorii, přičemž kategorie může mít několik kurzů. U kurzů eviduji název akce, místo konání, datum a čas začátku a ukončení akce, datum ukončení registrací, cenu kurzu a zálohu v poslední řadě můžeme zadat, jestli je kurz nějak věkově omezený. Ke každému kurzu můžeme také přidat doplňkové služby, u těch evidujeme název a cenu. Také je nutné ukládat informace o našich klientech. Pokud má zákazník zájem se zaregistrovat, vyžadujeme uživatelské jméno a heslo, které je ukládáno jako hash kvůli bezpečnosti. Jméno, příjmení, datum narození, email a telefonní číslo se vyžadují. Pokud je účastník mladší 18 let je také vyžadováno zadání rodiče. Ke každému uživateli potřebujeme také alespoň jednu adresu, více jich může být v případě, že fakturační adresa se liší od uživatelské adresy. Jako adresa účastníka ukládáme město, ulici, číslo popisné, poštovní směrovací číslo a stát. Pokud se jedná o fakturaci na firmu, ukládáme název společnosti, jméno a příjmení jednatele společnosti, identifikační číslo a daňové identifikační číslo.

V poslední řadě ukládáme informace o objednávkách. K tomu je potřeba ukládat o jakého uživatele se jedná, s kterou adresou je objednávka spárována, jaké kurzy obsahuje, a jestli jsou ke kurzům objednány také nějaké doplňkové služby. Pro každou objednávku generujeme unikátní název, který se skládá z aktuálního datumu a času vytvoření objednávky a identifikačního čísla v databázi. Také ukládáme cestu uložení faktury na serveru.

Jelikož tento modul je celkem rozsáhlý, rozhodl jsem se pro použití frameworku Yii.

Yii je volně šiřitelný, open-source framework pro tvorbu webových aplikací vyvinutý v PHP5 . Klade důraz na znouvopoužitelnost, jednoduchost, zefektivňuje vývoj aplikací a pomáhá zajistit rozšiřitelnost projektu.[6]

Jelikož byla potřeba generovat faktury v PDF souborech a následně je odesílat na email, zvolil jsem použití dvou rozšíření Yii frameworku. První je Yii-Pdf, který dokáže z html stránky vygenerovat PDF. Druhé rozšíření je SwiftMailer. Obě dvě rozšíření lze stáhnout zdarma na oficiálních stránkách Yii. Při vývoji v tomto frameworku jsem se naučil správně využívat principu MVC.

Yii framework obsahuje modul Gii který slouží k automatickému generování modelu, kontrolérů a pohledu. Nejprve vygenerujeme model, základně každý model dědí ze třídy CActiveRecord, která reprezentuje návrhový vzor ActiveRecord. ActiveRecord obaluje řádek v databázové tabulce, zapouzdřuje databázový přístup a přidává doménovou logiku. [7]

Následně přidáme základní informace, jako jsou název modelu, a tabulku v databázi, ze které si stáhne všechny atributy a vazby na ostatní tabulky se kterými má databázová tabulka vazby. Při generování kontrolérů se zadává název modelu, pro který bude vytvořen, poté zvolíme, z jaké třídy bude kontrolér dědit, základně je přednastavená třída Controller. Jako poslední zvolíme název akce, která se má v kontroleru vytvořit. Jako polední možnost je generování pohledu, zde zadáváme název modelu a kontroleru, který bude obsluhovat příslušné pohledy. Při generování se vytvoří pohledy pro zobrazení všech dat modelu, přidávání, editaci a mazání záznamů. Při vytváření se také edituje kontrolér, do kterého se přidávají metody pro zpracování dat.

Po vytvoření modelu bylo nutné také změnit rotování na takzvané pěkné adresy. To se nastavuje v konfiguračním souboru dopsáním url manažeru s nastavením urlFormat na path, a nastavení pravidel pro routování. I přesto vždy adresa musela začínat index.php/názevKontroleru. Abych tomuto zamezil, změnil jsem nastavení v souboru .htaccess kde je nutné nastavit rewriteEngine na hodnotu on a dopsat do rewriteRule regulární výraz, který nám zařizuje automatické přesměrování. Jen si musíme dát pozor, aby byl rewriteMod povolený v php.ini na servu.

Při vytváření přihlašovacího formuláře jsem narazil na problém jak umožnit uživateli přihlašovat se na více kurzů najednou. Rozhodl jsem se využít pluginu z knihovny jQuery UI který se nazývá choosen. Tento výborný plugin ovšem nesplňoval všechny požadavky, které byly potřeba, a proto jsem se rozhodl si plugin přizpůsobit mým požadavkům. V první řadě bylo důležité se v pluginu zorientovat, abych s ním dokázal pracovat. Po zorientování jsem implementoval funkci pro přidávání. Abych nemusel zbytečně načítat všechny doplňkové služby při načtení stránky a elementy skrývat, využil jsem technologie Ajax. Při každém přidání akce se pošle požadavek na server, který mi vrátí všechny doplňkové služby pro danou akci. Pokud zákazník kurz odstraní z vybraných, je potřeba všechny jeho doplňkové služby odstranit. Proto je nutné při přidávání elementu mu přiřadit jednoznačný identifikátor.

Yii framework obsahuje komponentu CWebUser která reprezentuje uživatele. Jelikož jsem potřeboval tuhle základní komponentu rozšířit, abych mohl evidovat i administrátory, musel jsem

vytvořit novou, vlastní komponentu, která dědí CWebUser, kde jsem si vytvořil metodu isAdmin která mi vrátí, jestli se jedná o administrátora či nikoliv. Pro přihlašování je Yii framework vyžaduje autentizaci, kterou vytvoříme tak, že vytvoříme instanci objektu UserIdentity.

#### Ukázka vytvoření instance objektu UserIdentity

```
$identity=new UserIdentity($record->user_name,$record->password);  
$duration= 3600*24*30; // 30 Dní  
Yii::app()->user->login($identity,$duration);
```

Po přihlášení jsem vytvořil uživatelské rozhraní, kde může uživatel zobrazovat si historické objednávky, upravovat svoje osobní údaje. V editaci uživatele lze přidávat další adresu, nebo smazat již zadanou. Jelikož by smazání z dat z databáze mohlo ovlivnit zobrazování historických dat, rozhodl jsem se nemazat data v databázi, ale přidat příznak jestli je adresa ještě aktuální.

## 4.2 Časová náročnost

**Tabulka 4.1:** Časová náročnost projektu

Fáze projektu	Časová náročnost
Analýza a návrh řešení	3 Dny
Instalace frameworku	1 Den
Přidávání doplňkových služeb	3 Dny
Komunikace se severem pomocí technologie Ajax	7 Dny
Generování PDF	3 Dny
Odesílání emailu	1 Dny
Testování aplikace a optimalizace	3 Dny

## 5 Nabyté znalosti

Při vykonávání odborné praxe ve společnosti SCOVECO s.r.o. jsem si osvojil a rozvinul znalosti získané při studiu na VŠB zejména navrhování databází, analyzování požadavků od zákazníků, tvorba diagramů. Především jsem uplatnil znalosti z programování a tvorbou webových aplikací v PHP, Javascript, jQuery. Seznámil se a naučil se správně aplikovat MVC architekturu a návrhového vzoru ActiveRecord. Také nemohu opomenout nabyté znalosti s verzovacím systémem Git, jehož znalost v dnešní době často patří k požadavkům firem.

Z pohledu konkrétních předmětů, které jsem absolvoval při studiu na VŠB, jsem využil znalosti z předmětů UDBS (Úvod do databázových systémů), DAIS (Databázové a informační systémy), PJ (Programovací jazyky), VIS (vývoj informačních systémů).



## 6 Závěr

Bakalářskou praxi jsem si zvolil, protože jsem chtěl vyzkoušet své dovednosti ve firemním prostředí, naučit se práce v týmu a získat zkušenosti, které bych mohl později uplatnit.

Při průběhu praxe jsem se naučil komunikovat a spolupracovat s členy týmu při práci na jednom projektu a využívat verzovacího nástroje Git, který práci zefektivňuje, zrychluje a zpříjemňuje. Dále jsem získal mnoho poznatků a zkušeností od svých kolegů programátorů.

# Literatura

- [1] [online]. [cit. 2014-04-22]. Dostupné z: <http://www.scoveco.cz/index.php/cz/o-spolecnosti/o-spolecnosti>
- [2] David Trowbridge, Dave Mancini. Enterprise Solution Patterns Using Microsoft.NET. 2003. Příručka společnosti Microsoft
- [3] FreeMarker [online]. [cit. 2014-04-26]. Dostupné z: <http://freemarker.org/>
- [4] Git [online]. [cit. 2014-04-26]. Dostupné z: <http://git-scm.com/>
- [5] Hibernate [online]. [cit. 2014-04-26]. Dostupné z: <http://hibernate.org/>
- [6] Yii [online]. [cit. 2014-04-24]. Dostupné z: <http://www.yiiframework.com/>
- [7] FOWLER, Martin. Patterns of enterprise application architecture. Boston: Addison-Wesley, November 05, 2002, xxiv, 533 s. The Addison-Wesley Signature Series. ISBN 978-0-321-12742-6.